What is Claimed is:

1. A method for representing at least one computer attack path in a network comprising:

receiving a starting point of a computer attack with respect to said network; and

generating a pruned augmented attack tree representing at least one attack path possible from said starting point, wherein, said starting point is a root of said pruned augmented attack tree, for a current node being evaluated as part of said generating, a resulting node and an edge connecting said current node to said resulting node are added to said pruned augmented attack tree if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node.

2. The method of Claim 1, wherein said pruned augmented attack tree is a tree including n levels, said starting point being a root of said tree at level 0, n being at least 0.

3. The method of Claim 2, wherein a node in said pruned augmented attack tree represents information about at least one of: an attacker state including a host and an attacker access level on said host, and a network state.

4. The method of Claim 3, wherein an edge from a first node at level x to a second node at level x+1 represents an action while in a first state including a first attacker state corresponding to said first node resulting in a second state including a second attacker state.

5. The method of Claim 4, wherein said action exploits a vulnerability on a host in said network.

6. The method of Claim 4, wherein said first attacker state represents a first host and a first attacker access level on said first host, and said second attacker state represents

at least one of: a second host and a second attacker access level on said second host, and said first host and a second attacker access level on said first host wherein said second attacker access level represents at least one of: an increase in attacker privilege, an increase in attacker access, and an increase in attacker knowledge.

7. The method of Claim 1, wherein said current node is at a level n, and said ancestors of said current node are located at levels in said pruned augmented attack tree at a level less than n.

8. The method of Claim 7, wherein said pruned augmented attack tree is generated using a breadth first search technique in which nodes are added to said pruned augmented attack tree at an nth level prior to adding any node from level n+1 to said pruned augmented attack tree.

9. The method of Claim 1, wherein a plurality of computer attack paths for said network are represented using a plurality of pruned augmented attack trees, each of said pruned augmented attack trees representing computer attack paths originating from a unique starting point.

10. The method of Claim 1, wherein said starting point is one of: from within said network and external to said network.

11. The method of Claim 6, further comprising:

evaluating each action that exploits a vulnerability of a host in accordance with connectivity data.

12. The method of Claim 11, wherein said connectivity data, said each action, and said vulnerability are stored in a database and determined prior to performing said generating.

13. The method of Claim 1, wherein, said pruned augmented attack tree has a property that a resulting node at a level "n+1" and an edge connecting a current node at level "n" to said resulting node are included in said pruned augmented attack tree if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node, said ancestor being a node at a level "x" < "n", and said instance of the resulting node being at level "x+1".

14. The method of Claim 1, further comprising:

determining which hosts in said network are equivalent forming a group; and representing said group with a single host.

15. The method of Claim 14, wherein a first host is equivalent to a second host if said first and second hosts have a same set of one or more vulnerabilities on a same set of one or more endpoints, said first and second hosts are not administrative hosts, said first and second hosts are not gateways, said first and second hosts have equivalent attack loss values; and said first and second hosts have equivalent connectivity.

16. The method of Claim 1, wherein said generating uses connectivity information, said connectivity information including a connection between two endpoints representing elements of a configuration of said network.

17. The method of Claim 16, wherein said connectivity information includes physical connectivity between network interfaces and logical connectivity through network communications protocols.

18. The method of Claim 16, wherein said connection is associated with a path including one or more hops.

19. The method of Claim 18, wherein each of said one or more hops is associated with at least one of: a filtering rule, a translation rule, and an interface of a host in said network.

20. The method of Claim 16, wherein at least one of said endpoints is associated with a vulnerability on said at least one endpoint.

21. The method of Claim 20, wherein said vulnerability has an associated action resulting in exploitation of said vulnerability.

22. The method of Claim 21, wherein said associated action is related to an entity representing at least one of: an attacker access level, attacker knowledge level, a change to a network state.

23. The method of Claim 1, wherein said pruned augmented attack tree is used to determine an effect of preventing at least one action.

24. The method of Claim 23, further comprising:

modifying said pruned augmented attack tree in accordance with eliminating at least one action in connection with a vulnerability associated with said host producing a modified augmented attack tree; and

evaluating said modified augmented attack tree.

25. The method of Claim 1, wherein connectivity data representing connectivity between pairs of endpoints in said network is used by said generating, and the method further comprising:

automatically generating said connectivity data in accordance with at least one translation rule, at least one filtering rule, and network configuration information.

26. The method of Claim 25, wherein said at least one translation rule includes at least one of: an address translation rule and a port translation rule.

27. The method of Claim 25, further comprising:

selecting at least one address of a starting point of a computer attack using at least one rule; and

determining a portion of said connectivity data using said at least one address.

28. The method of Claims 27, wherein said at least one rule includes at least one of a filtering rule and a translation rule.

29. The method of Claim 27, wherein said at least one address is used in said generating to represent an alternate connectivity of a host.

30. The method of Claim 27, wherein said address is one of an address in accordance with a communications protocol and an address associated with said network.

31. The method of Claim 5, further comprising:

using vulnerability data to determine at least one of: requirements for an action, an attacker state resulting from an action, and a network state resulting from an action, where said requirements include a locality describing whether a vulnerability can be exploited remotely over a network or locally on a host, said resulting attacker state includes an effect describing an access level or privilege or knowledge after an exploit of a vulnerability, and said resulting network state includes a denial of service describing a loss of service on a host after an exploit of a vulnerability.

32. A method for assessing security of a network comprising:

determining a network vulnerability score in accordance with first attack loss values for all hosts within said network that are compromised and second attack loss values associated with all hosts in said network.

33. The method of Claim 32, further comprising:

determining attack loss values for all hosts compromised in all attack trees of said network, each attack tree representing at least one computer attack path originating from a unique starting point.

34. The method of Claim 33, further comprising:

determining attack loss values for all hosts in said network.

35. The method of Claim 34, further comprising:

adding said attack loss values for all hosts compromised in all attack trees of said network producing a first sum;

adding said attack loss values for all hosts in said network producing a second sum; and

determining a ratio of said first sum to said second sum.

36. The method of Claim 32, further comprising:

receiving input for a proposed change to said network; and

assessing said proposed change to said network, said assessing including determining a revised value for said network vulnerability score.

37. The method of Claim 32, wherein said attack loss values are assigned to each host in said network in accordance with a value if said host is compromised.

38. The method of Claim 37, wherein said attack loss value for said each host is determined using at least one criteria selected from: criticality of data on said each host,

criticality of data available through said each host, criticality of a service available at said each host, and criticality of a service available through said each host.

39. The method of Claim 36, wherein said network vulnerability score is represented using a ratio of a first sum of attack loss values for all hosts compromised in one or more pruned attack trees to a second sum of attack loss values for all hosts in said network.

40. The method of Claim 33, wherein an attack tree vulnerability score is determined for each attack tree of said network and each attack tree vulnerability score is represented as a ratio of a first sum of attack loss values of all hosts compromised in said each attack tree to a second sum of attack loss values of all hosts in said network.

41. The method of Claim 39, further comprising:

determining a first network vulnerability score for said network without said proposed change;

determining a second network vulnerability score for said network with said proposed change; and

using said first and second network vulnerability scores in evaluating said proposed change with another proposed change.

42. The method of Claim 41, further comprising:

providing a prioritized set of one or more proposed changes to said network in accordance with a network vulnerability score associated with each of said one or more proposed changes.

43. The method of Claim 42, further comprising:

using one or more pruned attack trees in evaluating each of said one or more proposed changes.

44. The method of Claim 43, wherein one of said proposed changes includes eliminating at least one vulnerability at a host in said network.

45. The method of Claim 43, wherein all attack trees of said network are pruned augmented attack trees, each of said pruned augmented attack tree having a property that a resulting node at a level "n+1" and an edge connecting a current node at level "n" to said resulting node are included in said each pruned augmented attack tree if said edge and said resulting node are not already included in said each pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node, said ancestor being a node at a level "x" < "n", and said instance of the resulting node being at level "x+1".

46. A method of determining connectivity for a network using connectivity data comprising:

receiving at least one of: a rule set of one or more rules and information obtained from a network scanner;

determining at least one address of a host in said network using at least one of said rule set and said information obtained from a network scanner, said connectivity data including said at least one address of said host; and

determining connectivity between pairs of endpoints in said network using said connectivity data.

47. The method of Claim 46, further comprising:

determining whether connectivity is permitted in accordance with said rule set for a path between a pair of endpoints, said path including said host.

48. The method of Claim 46, wherein said rule set includes at least one of a filtering rule, a port translation rule, and an address translation rule.

49. The method of Claim 46, further comprising:

using said at least one address in modeling a change of a network address of a host after said host is compromised.

50. The method of Claim 49, wherein said connectivity data is used in generating an attack tree.

51. The method of Claim 50, wherein said at least one address is used in said generating to represent an alternate connectivity of a host.

52. The method of Claim 46, wherein said address is one of an address in accordance with a communications protocol and an address associated with said network.

53. The method of Claim 46, wherein said determining at least one address of a host further comprises:

examining at least one of: a filtering rule and an address translation rule;

determining an initial list of addresses;

sorting said initial list in accordance with address specificity wherein a first address in said initial list is more specific than a second address in said initial list if said first address specifies a smaller address range than said second address; and

forming a final list of addresses associated with said host, wherein, addresses in said initial list are examined in order from most to least specific, for each address in said initial list, said address is added to said final list if said each address is a specific address, for each address in said initial list that specifies a range an address is selected from the range for inclusion in the final list if the selected address is not already in the final list or included in a range of a more specific address in said initial list.

54. The method of Claim 46, wherein said determining connectivity comprises:

evaluating connectivity between a source endpoint and each target endpoint within a same subnet in said network; and

evaluating connectivity between a source endpoint and each target endpoint within other subnets in said network.

55. The method of Claim 46, further comprising:

evaluating connectivity between each pair of possible endpoints wherein all connectivity between a source and all possible target endpoints are explored prior to advancing to the next possible source endpoint.

56. A method for representing at least one computer attack path in a network comprising:

receiving a starting point of a computer attack with respect to said network; and

generating a data structure representing at least one attack path possible from said starting point, wherein, said starting point is a node in said data structure, and for a current node being evaluated as part of said generating, a resulting node and an edge connecting said current node to said resulting node are added to said data structure if said edge and said resulting node are not already included in said data structure with said edge connecting a predecessor of the current node to an instance of the resulting node, wherein said predecessor is a node along a path from the starting node to a node immediately preceding the current node.

57. The method of Claim 56, wherein said data structure is a representation of at least one of: an augmented rooted tree, a non augmented rooted tree, a free tree, a directed acyclic graph, an undirected graph, a graph and a tree.

58. The method of Claim 57, wherein said data structure includes at least one of: an array, a linked list, a hash table, an adjacency list, and an adjacency matrix.

59. A computer program product for representing at least one computer attack path in a network comprising executable code that:

receives a starting point of a computer attack with respect to said network; and

generates a pruned augmented attack tree representing at least one attack path possible from said starting point, wherein, said starting point is a root of said pruned augmented attack tree, and for a current node being evaluated, a resulting node and an edge connecting said current node to said resulting node are added to said pruned augmented attack tree if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node.

60. The computer program product of Claim 59, wherein said pruned augmented attack tree is a tree including n levels, said starting point being a root of said tree at level 0, n being at least 0.

61. The computer program product of Claim 60, wherein a node in said pruned augmented attack tree represents information about at least one of: an attacker state including a host and an attacker access level on said host, and a network state.

62. The computer program product of Claim 61, wherein an edge from a first node at level x to a second node at level x+1 represents an action while in a first state including a first attacker state corresponding to said first node resulting in a second state including a second attacker state.

63. The computer program product of Claim 62, wherein said action exploits a vulnerability on a host in said network.

64. The computer program product of Claim 62, wherein said first attacker state represents a first host and a first attacker access level on said first host, and said second attacker state represents at least one of: a second host and a second attacker access level

on said second host, and said first host and a second attacker access level on said first host wherein said second attacker access level represents at least one of: an increase in attacker privilege, an increase in attacker access, and an increase in attacker knowledge.

65. The computer program product of Claim 59, wherein said current node is at a level n, and said ancestors of said current node are located at levels in said pruned augmented attack tree at a level less than n.

66. The computer program product of Claim 65, further comprising executable code that generates said pruned augmented attack tree using a breadth first search technique in which nodes are added to said pruned augmented attack tree at an nth level prior to adding any node from level n+1 to said pruned augmented attack tree.

67. The computer program product of Claim 59, wherein a plurality of computer attack paths for said network are represented using a plurality of pruned augmented attack trees, each of said pruned augmented attack trees representing computer attack paths originating from a unique starting point.

68. The computer program product of Claim 59, wherein said starting point is one of: from within said network and external to said network.

69. The computer program product of Claim 64, further comprising:

executable code that evaluates each action that exploits a vulnerability of a host in accordance with connectivity data.

70. The computer program product of Claim 69, further comprising executable code that stores said connectivity data, said each action, and said vulnerability in a database prior to generating said pruned augmented attack tree.

71. The computer program product of Claim 59, wherein, said pruned augmented attack tree has a property that a resulting node at a level "n+1" and an edge connecting a current node at level "n" to said resulting node are included in said pruned augmented attack tree if said edge and said resulting node are not already included in said pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node, said ancestor being a node at a level "x" < "n", and said instance of the resulting node being at level "x+1".

72. The computer program product of Claim 59, further comprising executable code that:

determines which hosts in said network are equivalent forming a group; and

represents said group with a single host.

73. The computer program product of Claim 72, wherein a first host is equivalent to a second host if said first and second hosts have a same set of one or more vulnerabilities on a same set of one or more endpoints, said first and second hosts are not administrative hosts, said first and second hosts are not gateways, said first and second hosts have equivalent attack loss values; and said first and second hosts have equivalent connectivity.

74. The computer program product of Claim 59, wherein said executable code that generates said pruned augmented attack tree uses connectivity information, said connectivity information including a connection between two endpoints representing elements of a configuration of said network.

75. The computer program product of Claim 74, wherein said connectivity information includes physical connectivity between network interfaces and logical connectivity through network communications protocols.

76. The computer program product of Claim 74, wherein said connection is associated with a path including one or more hops.

77. The computer program product of Claim 76, wherein each of said one or more hops is associated with at least one of: a filtering rule, a translation rule, and an interface of a host in said network.

78. The computer program product of Claim 74, wherein at least one of said endpoints is associated with a vulnerability on said at least one endpoint.

79. The computer program product of Claim 78, wherein said vulnerability has an associated action resulting in exploitation of said vulnerability.

80. The computer program product of Claim 79, wherein said associated action is related to an entity representing at least one of: an attacker access level, attacker knowledge level, a change to a network state.

81. The computer program product of Claim 59, further comprising executable code that uses said pruned augmented attack tree to determine an effect of preventing at least one action.

82. The computer program product of Claim 81, further comprising executable code that:

modifies said pruned augmented attack tree in accordance with eliminating at least one action in connection with a vulnerability associated with said host producing a modified augmented attack tree; and

evaluates said modified augmented attack tree.

83. The computer program product of Claim 59, wherein connectivity data representing connectivity between pairs of endpoints in said network is used by said

executable code that generates, and the computer program product further comprising executable code that:

automatically generates said connectivity data in accordance with at least one translation rule, at least one filtering rule, and network configuration information.

84. The computer program product of Claim 83, wherein said at least one translation rule includes at least one of: an address translation rule and a port translation rule.

85. The computer program product of Claim 83, further comprising executable code that:

selects at least one address of a starting point of a computer attack using at least one rule; and

determines a portion of said connectivity data using said at least one address.

86. The computer program product of Claim 85, wherein said at least one rule includes at least one of a filtering rule and a translation rule.

87. The computer program product of Claim 85, wherein said at least one address is used in said generating to represent an alternate connectivity of a host.

88. The computer program product of Claim 85, wherein said address is one of an address in accordance with a communications protocol and an address associated with said network.

89. The computer program product of Claim 63, further comprising:

executable code that uses vulnerability data to determine at least one of: requirements for an action, an attacker state resulting from an action, and a network state resulting from an action, where said requirements include a locality describing whether a

vulnerability can be exploited remotely over a network or locally on a host, said resulting attacker state includes an effect describing an access level or privilege or knowledge after an exploit of a vulnerability, and said resulting network state includes a denial of service describing a loss of service on a host after an exploit of a vulnerability.

90. A computer program product that assesses security of a network comprising: executable code that determines a network vulnerability score in accordance with first attack loss values for all hosts within said network that are compromised and second attack loss values associated with all hosts in said network.

91. The computer program product of Claim 90, further comprising:

executable code that determines attack loss values for all hosts compromised in all attack trees of said network, each attack tree representing at least one computer attack path originating from a unique starting point.

92. The computer program product of Claim 91, further comprising:

executable code that determines attack loss values for all hosts in said network.

93. The computer program product of Claim 92, further comprising executable code that:

adds said attack loss values for all hosts compromised in all attack trees of said network producing a first sum;

adds said attack loss values for all hosts in said network producing a second sum; and

determines a ratio of said first sum to said second sum.

94. The computer program product of Claim 90, further comprising executable code that:

receives input for a proposed change to said network; and

assesses said proposed change to said network, said executable code that assesses including executable code that determines a revised value for said network vulnerability score.

95. The computer program product of Claim 90, wherein said attack loss values are assigned to each host in said network in accordance with a value if said host is compromised.

96. The computer program product of Claim 95, wherein said attack loss value for said each host is determined using at least one criteria selected from: criticality of data on said each host, criticality of data available through said each host, criticality of a service available at said each host, and criticality of a service available through said each host.

97. The computer program product of Claim 94, wherein said network vulnerability score is represented using a ratio of a first sum of attack loss values for all hosts compromised in one or more pruned attack trees to a second sum of attack loss values for all hosts in said network.

98. The computer program product of Claim 91, wherein an attack tree vulnerability score is determined for each attack tree of said network and each attack tree vulnerability score is represented as a ratio of a first sum of attack loss values of all hosts compromised in said each attack tree to a second sum of attack loss values of all hosts in said network.

99. The computer program product of Claim 97, further comprising executable code that:

determines a first network vulnerability score for said network without said proposed change;

determines a second network vulnerability score for said network with said proposed change; and

uses said first and second network vulnerability scores in evaluating said proposed change with another proposed change.

100. The computer program product of Claim 99, further comprising:

executable code that provides a prioritized set of one or more proposed changes to said network in accordance with a network vulnerability score associated with each of said one or more proposed changes.

101. The computer program product of Claim 100, further comprising:

executable code that uses one or more pruned attack trees in evaluating each of said one or more proposed changes.

102. The computer program product of Claim 101, wherein one of said proposed changes includes eliminating at least one vulnerability at a host in said network.

103. The computer program product of Claim 101, wherein all attack trees of said network are pruned augmented attack trees, each of said pruned augmented attack tree having a property that a resulting node at a level "n+1" and an edge connecting a current node at level "n" to said resulting node are included in said each pruned augmented attack tree if said edge and said resulting node are not already included in said each pruned augmented attack tree with said edge connecting an ancestor of the current node to an instance of the resulting node, said ancestor being a node at a level "x" < "n", and said instance of the resulting node being at level "x+1".

104. A computer program product that determines connectivity for a network using connectivity data comprising executable code that:

receives at least one of: a rule set of one or more rules and information obtained from a network scanner;

determines at least one address of a host in said network using at least one of said rule set and said information obtained from a network scanner, said connectivity data including said at least one address of said host; and

determines connectivity between pairs of endpoints in said network using said connectivity data.

105. The computer program product of Claim 104, further comprising:

executable code that determines whether connectivity is permitted in accordance with said rule set for a path between a pair of endpoints, said path including said host.

106. The computer program product of Claim 104, wherein said rule set includes at least one of a filtering rule, a port translation rule, and an address translation rule.

107. The computer program product of Claim 104, further comprising:

executable code that uses said at least one address in modeling a change of a network address of a host after said host is compromised.

108. The computer program product of Claim 107, wherein said connectivity data is used by executable code that generates an attack tree.

109. The computer program product of Claim 108, wherein said at least one address is used in said generating to represent an alternate connectivity of a host.

110. The computer program product of Claim 104, wherein said address is one of an address in accordance with a communications protocol and an address associated with said network.

111. The computer program product of Claim 104, wherein said executable code that determines at least one address of a host further comprises executable code that:

examines at least one of: a filtering rule and an address translation rule;

determines an initial list of addresses;

sorts said initial list in accordance with address specificity wherein a first address in said initial list is more specific than a second address in said initial list if said first address specifies a smaller address range than said second address; and

forms a final list of addresses associated with said host, wherein, addresses in said initial list are examined in order from most to least specific, for each address in said initial list, said address is added to said final list if said each address is a specific address, for each address in said initial list that specifies a range an address is selected from the range for inclusion in the final list if the selected address is not already in the final list or included in a range of a more specific address in said initial list.

112. The computer program product of Claim 104, wherein said executable code that determines connectivity comprises executable code that:

evaluates connectivity between a source endpoint and each target endpoint within a same subnet in said network; and

evaluates connectivity between a source endpoint and each target endpoint within other subnets in said network.

113. The computer program product of Claim 104, further comprising:

executable code that evaluates connectivity between each pair of possible endpoints wherein all connectivity between a source and all possible target endpoints are explored prior to advancing to the next possible source endpoint.

114. A computer program product that represents at least one computer attack path in a network comprising executable code that:

receives a starting point of a computer attack with respect to said network; and

generates a data structure representing at least one attack path possible from said starting point, wherein, said starting point is a node in said data structure, and for a current node being evaluated as part of said generating, a resulting node and an edge connecting said current node to said resulting node are added to said data structure if said

edge and said resulting node are not already included in said data structure with said edge connecting a predecessor of the current node to an instance of the resulting node, wherein said predecessor is a node along a path from the starting node to a node immediately preceding the current node.

115. The computer program product of Claim 114, wherein said data structure is a representation of at least one of: an augmented rooted tree, a non augmented rooted tree, a free tree, a directed acyclic graph, an undirected graph, a graph and a tree.

116. The computer program product of Claim 115, wherein said data structure includes at least one of: an array, a linked list, a hash table, an adjacency list, and an adjacency matrix.